DOCUMENT RESUME

ED 138 286 IR 004 715

AUTHOR Pelczarski, Mark
TITLE INFO--A Data Manipulation and Retrieval System in BASIC. Illinois Series on Educational Application of Computers. Number 9.
PUB DATE May 76
NOTE 29p.; For related documents, see IR 004 716 and IR 004 719 ; Some appendices may be marginally legible due to small print size of the original

EDRS PRICE MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS *Computer Programs; *Data Processing; *Information Retrieval; Input Output; Programing
IDENTIFIERS *BASIC; DEC 10; INFO

ABSTRACT

INFO is a group of programs, written in BASIC on the DEC 10 system, that allow a wide range of uses for data files, easy and versatile editing of these files, and different means of outputting data. This report discusses the development of INFO, its uses in manipulating files, recommended changes, and possible applications. Appendices provide examples of output format. (DAG)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED138286

U S DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Number 9

The Illinois Series on

Educational Applications of Computers

INFO - A Data Manipulation and Retrieval System in BASIC

PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY
Bruce Hicks
TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER

Mark Pelczarski

BEST COPY AVAILABLE

May 1976

Department of Secondary Education
College of Education
University of Illinois
Urbana, Illinois

© Bruce Hicks (1976)

IR004715

Preface

Although several earlier ISEAC papers have treated certain aspects of administrative applications of computers, this is the first paper devoted entirely to such applications. The paper describes the functions and mode of use of a system written in BASIC for management of small and medium sized data bases. The system's flexibility and easy use recommends it for many different applications in schools whether by administrators, teachers or students.

The Editors plan to publish two more papers in the Series that are concerned primarily with administration or management aspects of computer applications. Number 10 will describe procedures and files for documentation and management of the course-ware for a laboratory or department devoted to educational applications of computers. Number 11 will describe a computer-assisted instruction management program and its applications.

Bruce L. Hicks

TABLE OF CONTENTS

I. Introduction

II. Development

III. Use of INFO - Manipulation of files

3.1 Creating a file
3.2 Signing in on INFO
3.3 CATALOG
3.4 ADD
3.5 ALTER
3.6 DELETE
3.7 HELP

IV. Use of INFOPR - column output of data

V. Use of INFOAD - heading manipulation

5.1 Change of password
5.2 Adding a heading
5.3 Changing a heading and/or corresponding data
5.4 Deleting a heading

VI. INFO - the programs

6.1 Data files
6.2 Character strings
6.3 Other conversion help
6.4 Recommended changes and additions

VII. Possible Uses

Appendix A - Output from INFO
Appendix B - Output from INFOPR
Appendix C - Help section
Appendix D - Listing
Appendix E - Definitions of variables

ISEAC Number 9

INFO--A Data Manipulation and Retrieval System in BASIC

Mark Pelczarski

I. Introduction

INFO is a group of programs, written in BASIC, that allow a wide range of uses for data files, easy and versatile editing of these files, and different means of outputting data.* It was written to handle small needs in file size and is still untested with large files. Files of third order size (1000 pieces of information) should be suitable for INFO; fourth order files (10,000 pieces of data) should be considered skeptically. The largest file used to date has been about 50 by 10--500 pieces of data.

INFO was written on a DECsystem 10 time-shared computer, in Digital Equipment Corporation's version of BASIC. Data file manipulation and access time on some other computers may make INFO difficult to use. (See section 6.1 of this paper for a description of DEC 10's data file handling in INFO). Questions about INFO and the implementation of INFO can be directed to Mark Pelczarski, c/o Department of Secondary Education, University of Illinois, 398 Education, Urbana, Illinois 61801.

II. Development

INFO was originally written in Spring 1975 as a specialized data handling program that would allow storage of information about other programs available on the DEC 10, and provide easy access and alterations of this data. The following information was kept for each program entered in the file: Program Name, Programmer, Language, Length, a one-line description, and the date of the last edit of this information. The following fall a similar program was needed to keep graduate student records. It was apparent that there were several possible applications

---

* A number of possible uses of INFO in education are listed in Section VII.

for a program of the same form as INFO, so rather than write a new program each time, it was decided that INFO should be generalized. The new program would allow different headings for different uses, and allow the number of headings to vary according to the needs of each particular use. One file could be created with the headings NAME and PHONE #, and another could be created with such headings as NAME, TYPE OF FINANCIAL AID, LAST DEGREE RECEIVED, and 1ST SEMESTER ENROLLED, and each could use the generalized structure of the INFO program. By combining several uses into one general structure, programming time and computer space would be saved in the future.

INFO allows adding data to a file, altering existing data, deleting existing data, and output of data (individually by entry name, all data in roster form accompanied by headings, or a list of entry names). (See Appendix A.) Two supplementary programs, INFOPR and INFOAD, were written to diversify possible uses of INFO. INFOPR allows column output of data (Appendix B), and allows multiple copies of such output. INFOAD makes it possible to add new headings to an existing file, to delete headings and their corresponding data from a file, to change headings and/or their corresponding data for the entire file, or to change an optional password. Another supplementary program being planned would allow searches for particular information under certain headings, and possibly a sort that could alphabetize data under headings other than the first (which is NAME, and already alphabetized), or sort numerical data.

III. Use of INFO - Manipulation of files

3.1 Creating a new file*

To create a new data file for the INFO program, it is necessary to set up three new files and alter another. As an illustration, I will use the name "FILE" for the new data file to be set up. The name would usually be chosen to have some

---

* This section is optional. It should be read when a new data file is to be created.

meaning;(for example "GRADS" for graduate student records or "PROGS" for programs) This file is the data file in which specifications and the column headings are kept. The actual data will be stored in FILEF and FILEG (for GRADS, GRADSF and GRADSG; for PROGS, PROGSF and PROGSG). The names are determined by adding "F" and a "G" to your file name, therefore your filename must be one character less than the maximum allowed by the computer (six on the DEC 10).

To create FILEF and FILEG it is just necessary to inform the computer that you will be using these names. On the DEC 10 this is done by the following commands:

```
NEW FILEF
SAVE
```

This tells the computer tnat FILEF is a new name, and that you wish to save it. The same is done for FILEG.

Since FILE will contain specifications, it must be set up slightly differently. FILE is a line-numbered, sequential access file (for full explanation see Section 6.1).

The first entry in FILE is the password which will be required to access FILE when using the INFO programs. The password should be one word, enclosed in quotes. If a password is not desired, two double quotes side by side ("") should be used. This is called a null string, and INFO checks for it before asking for the password.

The second entry is the maximum length you are allowing for any piece of data to be stored. The DEC 10 maximum is 132, but this varies with the machine. The advantage of setting the maximum to be smaller is that it is used in INFOPR when computing the spacing between columns. Since computer paper is at most 132 characters wide, if one wanted five columns of output, the maximum lengths should be set less than 25 (allowing one space separation between columns).

The third entry in FILE is a brief description (approximately 40 characters or less) of what your file is (e.g. "GRADUATE STUDENT RECORDS"). It should be enclosed in double quotes.

The fourth entry is the first heading you wish to use for your file (usually "NAME"). The following entries will be the rest of your headings.

Here is an example of setting up FILE on the DEC 10:

```
NEW FILE
10 "PASSWORD"
20 "20"
30 "DESCRIPTION OF PURPOSE"
40 "NAME"
50 "SECOND HEADING"
60 "THIRD HEADING"
70 "LAST HEADING"
SAVE
```

The numbers before the lines are the line numbers mentioned earlier, and are only used in editing. If you had made a mistake in line 40 and discovered it after you had finished the next line, you could have typed 40 again after line 50 and it would correctly replace the old line 40 and be inserted before 50.

The file that must be altered when creating a new INFO file is a file called INFOLS, which stands for INFO List. This file is simply a list of INFO data files such as "GRADS" and "PROGS". It is also a line numbered file, and since it already exists when you are creating your file "FILE", you would refer to it as follows (on the DEC 10):

```
OLD INFOLS
LIST
```

The LIST command causes the contents of INFOLS to be printed. If file "GRADS" and "PROGS" already exist, the computer response would be similar to:

```
10 "GRADS"
20 "PROGS"
```

To add "FILE", you would type:

```
30 "FILE"
REPLACE
```

Which adds line 30 and then replaces the old version (lines 10 & 20) with the new one (lines 10, 20 & 30).

You are now ready to use INFO.

3.2 Signing in on INFO

When running INFO, the first response you get is:

```
WHICH FILE DO YOU WANT?  (TYPE 'NONE' FOR NONE)
CHOICES:
GRADS       PROGS       FILE?
```

where GRADS, PROGS, and FILE are the three files in INFOLS. You respond by typing your choice and pressing the RETURN key. If you have a password the computer responds:

```
PASSWORD?
```

After typing in your password (you get two tries) the computer asks if you want Long Form Instructions. These instructions are identical to the HELP section described later and are listed in Appendix C. After typing 'YES' or 'NO', and after the HELP Section if you chose YES, the computer prints the following:

```
CHOICES:  1)CATALOG,   2)ADD,   3)ALTER,   4)DELETE,   5)EXIT,   6)HELP
?
```

Type in the number of your choice and RETURN.

3.3 CATALOG

If you pick choice 1, the computer responds:

```
CHOICES:  (1) LIST OF ENTRY NAMES, (2) DATA ON ONE ENTRY, (3) DATA ON ALL ENTRIES,
          (4) NONE

?
```

Choice 1 gives a list of first data items for each entry (the first item is usually "NAME"; see Appendix A). If you want to refer later to a specific entry in the file, you reference it by this first data item. (See Appendix A, where, for example, the name "BROWN/JAMES" references the other six items, from "DURBIN" to "4/10/76 - MP", in this entry.)

Choice 3 gives all the data entered into the file, grouped by entry with each item accompanied by its headings (See Appendix A). Choice 2 allows you to get the same type of output for any particular entry. The computer responds:

NAME OF ENTRY?

after this you would type the name (first data item) of the entry you want, or any unique beginning of that name. (For example, "JOHNSON" may be abbreviated "J" if no other name starts with "J", but if there is also a "JONES", a minimum of "JOH" would be necessary for "JOHNSON". If you make an error, a message is given and you try again.)

Choice 4 takes you back to the original choices of

CATALOG, ADD, ALTER, DELETE, EXIT, or HELP.

3.4 ADD

When adding data, the computer will print out the headings for the file and wait for the appropriate data after each. The new data will be inserted in the file alphabetically (or numerically) according to the first data item. If a particular data item is unknown, it may be skipped by leaving a blank (pressing -RETURN-). None of the data should include commas due to the way character strings are handled. Following is an example of adding data. The underlined portion is what the user typed:

```
DO NOT USE COMMAS IN ANY ANSWERS
MAXIMUM LENGTH OF LINE SHOWN BELOW
--------------------------------!
NAME
?DOE/JOHN                          </ is used rather than ,>
STREET ADDRESS
?4436 MAIN
CITY/STATE
?URBANA/ILL.
PHONE
?                                  <not known, so left blank>
LAST UPDATED
?4-20-76                           <today's date>
```

3.5 ALTER

After choice 3, the computer responds:

```
WHICH ENTRY DO YOU WANT TO CHANGE ('NONE' FOR NONE)
?
```

You respond with the entry name (or any unique beginning). If you type 'NONE' you get the original choices back again.

The computer responds to a correct name with:

```
AFTER EACH LINE, TYPE 'S' TO SAVE, 'C' TO CHANGE
```

Using the example just added, here's a sample of what would follow. The underlined portion is what the user typed.

```
NAME:  DOE/JOHN
STREET ADDRESS:  4436 MAIN ? S
CITY/STATE:  URBANA/ILL.   ? S
PHONE:                     ? C
CHANGE TO?  367-1829
LAST UPDATED:  4-20-76     ? C
CHANGE TO?  4-23-76
```

The program then returns to the original choices.

3.6 DELETE

Choice 4 responds with:

```
WHICH ENTRY DO YOU WANT DELETED FROM LIST ('NONE' FOR NONE)
?
```

A correct entry name or unique beginning (e.g. DOE for DOE/JOHN) is followed by:

```
DELETE DOE/JOHN?  ('YES' or 'NO')?
```

which is to verify that you do want that entry deleted. All data under that entry will be deleted. 'Y' or 'YES' is followed by:

```
DOE/JOHN DELETED FROM LIST
```

and 'N' or 'NO' is followed by:

```
DOE/JOHN NOT DELETED
```

The program then returns to the original choices.

3.7 HELP

This choice prints out instructions, which are the same as the long form instructions mentioned earlier. They are listed in Appendix C.

IV. Use of INFOPR

INFOPR was written to be used on the Decwriter Terminals available with our system which use full 132 column output paper rather than 80 column teletype paper. They are interactive terminals similar to a teletype, but faster. INFOPR is short and can easily be adapted for use on a standard teletype or a line printer. The sign-in part of the program is identical to that of INFO. After choosing the file and giving the password, if there is one, the user is asked how many copies he wants. He may choose between 0 and 50. (The maximum of 50 may be changed by minor program alteration.) The next choice is whether the output is to be single spaced or double spaced. After this decision, the computer responds:

```
SET THE PAPER TO THE TOP OF THE NEXT PAGE NOW
THEN PRESS -RETURN-
```

Setting the paper to the top of the page causes the output to be such that separate copies may be torn off at the perforations rather than cut apart. INFOPR divides the page into equal length columns, so if a piece of data overruns its column it is chopped off at that point so the next column can begin. A sample of INFOPR column output is in Appendix B.

V. Use of INFOAD

The sign-on procedure for INFOAD is identical to INFO and INFOPR. After a correct sign on, the user receives the following:

```
CHOICES:  (1) CHANGE PASSWORD
          (2) ADD A NEW HEADING
          (3) CHANGE A HEADING (OR CHANGE ALL THE DATA UNDER A PARTICULAR
              HEADING.  TO DO THIS, CHANGE THE HEADING TO ITSELF, THEN
              CHANGE THE DATA.)
          (4) DELETE A HEADING
          (5) EXIT THE PROGRAM

?
```

5.1 Change of password

If a password for the file under which you signed in does not exist (i.e. the password given in the heading file is the null string:""), choice 1 yields the response YOU CANNOT CHANGE A NON-EXISTENT PASSWORD. If the file does have a password, INFOAD asks for the new password, then has you type it once over to verify that it was correct the first time. If the two do not match, the program has the user start again; when they do match the password is changed.

5.2 Adding a heading

INFOAD allows one to add a new heading to a file between any of the existing headings, or after the last one. INFOAD allows the insertion of new data for that heading in the proper places in the data file, or inserts blanks as data in the proper spots, allowing the user to change them at another time. The first response after choosing to add is.

```
CHOICES:
     (1) INITIALIZE ALL DATA FOR THE NEW HEADING NOW
     (2) HAVE THE DATA FOR THE NEW HEADING SET BLANK AND INITIALIZE LATER

?
```

After you decide, the computer will give a list of the current headings, numbered from 1 to the number of headings, followed by:

```
TYPE THE NUMBER OF THE HEADING AFTER WHICH YOU WANT TO INSERT THE NEW HEADING
(TYPE '0' TO NOT INSERT)
?
```

After you pick the location of the new heading:

```
WHAT IS THE NEW HEADING (IF YOU WANT A COMMA  IN IT, PLEASE ENCLOSE THE HEADING
IN DOUBLE QUOTES)
?
```

If you had chosen to set all data blank for the time being, after typing in the heading the computer would insert all necessary blanks and you would be done. If you chose to initialize data, after giving the new heading you would be given the entry name (first data item) of each entry in your file. Each name would be followed by your new heading and a question mark, at which you type in the appropriate data. After you have done this for each entry you are finished.

5.3 Changing a heading and/or corresponding data

The first response is:

```
CHOICES:
     (1) CHANGE CORRESPONDING DATA ALSO
     (2) LEAVE DATA ALONE
?
```

The first choice lets the user change all the data under the heading he picks, therefore if he "changes" the heading to itself he may change all data under an existing heading. If the heading is changed to something different, the new data replaces the old. The second choice changes heading only. After a choice is made the computer lists the current headings, each preceded by a number. The user is asked the number of the heading he wished to change, and then asked for the new heading. After it is given the computer replaces the old heading, then lists each entry name followed by the new heading and a question mark, if the corresponding data is to be changed.

5.4 Deleting a heading

The computer will list the current headings, each numbered, and respond:

```
TYPE THE NUMBER OF THE HEADING YOU WANT DELETED, OR TYPE '0' FOR NONE (DATA
  UNDER THAT HEADING WILL BE LOST ALSO)?
```

After choosing the number, the heading is printed and the user is asked to verify that he wants it deleted. If he chooses "YES", the heading is deleted.

VI. INFO - the programs

The first concern of those who would want to use this program should be "Will it work on our computer?" The answer depends upon how much modification of INFO is necessary and upon personnel to make the changes. If it is used on a DEC 10 with BASIC, little or no work would be necessary. Much work would be involved if it is used on a computer that does not support the BASIC language. Translation to another language should be done using only the framework of the program, since other languages may allow more efficient methods. These are the extremes, and chances are that most conversions will be to different versions of BASIC. The differences that would most affect translation from one BASIC to another would be in character string handling and data file manipulation.

6.1 DATA FILES

There are two main types of data files used with BASIC on the DEC 10: random access files and sequential access files. Sequential access files have two types, non-line numbered and line numbered. Random access files are divided into equal length data items; therefore the computer is able to access any specific data item by knowing where in the list it is (e.g. if the data items are 10 characters long and the one you want is 23rd in the list, the computer goes directly to the 221st through 230th characters). Sequential access files allow varying length for data items and take up much less space. Whereas with a random access file you must declare a maximum length, thus wasting space with every item shorter than that length, sequential access files allow you to pack data as closely as possible. The drawback with sequential access files is the necessity to access everything sequentially; if you want the 23rd data item you must first read the 22 items before it. Worse, if you then want to read the 22nd item again, you must

also read the first 21 again. When writing into a sequential access file you must read and store the entire file elsewhere, scratch the file, and then start writing into it from the beginning. INFO makes the best use of these two types of files by storing the data in a sequential access file (FILEF) and copying that file into a random access file (FILEG) at the beginning of the program. When the user is done FILEG is copied back into FILEF. Random access files are declared with the statement FILE:1, "NAME". Sequential access files are declared with FILE #1, "NAME".

With sequential access files, line-numbered files allow editing without use of a program. INFOLS and the heading file (FILE) are both line numbered and are both altered without use of a program. The distinction the computer makes is that line numbered sequential access files use READ and WRITE, and non-line numbered use INPUT and PRINT. Examples of statements for line numbered files are READ #1, A$ and WRITE #5, M$. Examples for non-line numbered are INPUT #2, N$ and PRINT #2, A$. Random access files make no distinction between PRINT and WRITE, or INPUT and READ. PRINT:5, A$ serves the same purpose as WRITE:5, A$.

Special file statements on the DEC 10 are:

QUOTE #1 - causes all data in file #1 to be treated as character strings.

IF END #2, THEN 500 - causes transfer to line 500 if the
and
IF END:2, THEN 500 - pointer to the specified data file has reached just beyond the last item.

SET:1, 23 - sets the pointer of a random access file to the 23rd data item. (A variable can be used in place of 23 in the example.)

LOC(:1) - a function that returns the location of the pointer of a random access file.

LOF(:1) - a function that returns the location of the last item of a random access file.

In the three INFO programs the convention for file numbers has been as follows:

:1 INFO - random access file FILEG, where data is kept and manipulations done.

#1 INFOPR, INFOAD - sequential access file FILEF, where data is stored.

#2 INFO - sequential access file FILEF, which holds data and is transferred to :1.

#3 INFO, INFOPR, INFOAD - sequential access line-numbered file containing headings (FILE).

#4 INFO, INFOPR, INFOAD - sequential access, line-numbered file INFOLS, containing names of all heading files.

:5 INFO, INFOAD - random access, scratch file INFOSC used for manipulations.

6.2 Character Strings

The following three character functions are used in the INFO programs:

VAL(N$) - If N$ is the character representation of a number, VAL(N$) gives the numerical value. It is used when reading the maximum data length from the heading file, which is QUOTE'd.

LEN(N$) - Gives the number of characters in the string N$. It is used to check whether a data item that is input is longer than the maximum allowed.

LEFT$(N$,3) - The one most often used in INFO. This statement gives the first 3 characters of N$ (the 3 leftmost characters). It is used to chop off data items that are longer than the specified maximum to judge Yes/No answers, and to permit unique abbreviations to reference names. Yes/No answers are judged only on the basis of the first character - 'Y' or 'N'. When searching for a match to an

abbreviation, first the length of the abbreviation is determined using LEN, then the abbreviation is compared to chopped versions of entry names using LEFTS. (A variable may be used in place of 3 in the sample.)

6.3 Other conversion help

Since not all computer paper has pages of the same size, and INFOPR has to be able to page, there is a DATA statement at the end of INFOPR specifying the number of lines per page. This was made necessary due to the suspension of the PAGE command on our DEC 10.

Listings of all three programs are in Appendix D. These are complete as of April 29, 1976.

A list of variable names and definitions for each program appears in Appendix E. This list is virtually essential to program modification since Remark statements in the program do not define variables. Remark statements are not executed, but they take more disk space than other statements, and therefore were not used extensively.

6.4 Recommended changes and additions

If modifications are to be made to the INFO package, the following would be the most useful to pursue:

1) A major restructing of the data file arrangement would speed up execution time considerably if large files are to be used. Data is now stored sequentially, and new entries are inserted in the alphabetical spot where they belong, causing much data to be shifted. A linked list structure would eliminate this necessity.
2) A small program that guides the user through the initial creation of the data files (section 3.1) would be relatively easy to write.

3) Another useful program would be one that allows a search through the data for items that have certain characteristics. It would be easiest to search through the data and put each entry that meets desired specifications into INFOSC (the scratch file) and once the data is searched, chain to INFOPR at a location set aside for printing the contents of INFOSC.

4) A third useful program would be similar to the search, except it would sort data by items other than the first heading. These can be either numeric or character strings.

VII Possible Uses

Following are some possible applications of INFO as suggested by various people who have used it. The list is far from complete, but is meant to give a good cross section of uses.

1) Maintaining student records (class or administrative).
2) Keeping a list of resources (people, books, programs...).
3) Keeping records of class or school projects.
4) Setting up and maintaining data files used by other computer programs.
5) Storing survey results.
6) Indexes of information by subject area.
7) Lists of colleges and enrollment data for guidance counselling.

----- APPENDIX A: OUTPUT FROM INFO -----

```
**********

BROWN/JAMES
COGSWELL/GAYLA
JACKSON/JUDITH
LECKNER/MICHAEL
NORRIE/TERRI
O'BRIEN/DON
STORM/JEFF

**********
END OF CATALOG

CHOICES: (1) LIST OF ENTRY NAMES, (2) DATA ON ONE ENTRY,
         (3) DATA ON ALL ENTRIES, (4) NONE
 ?3

NAME:          BROWN/JAMES
ADVISOR:       DURBIN
AID:           NONE
STATUS:        ON CAMPUS
DEG.PROG.:     M.A.
DEGREE:        B.S.
UPDATED/BY:    4/10/76-MP

NAME:          COGSWELL/GAYLA
ADVISOR:       SJOKEN
AID:           NONE
STATUS:        SUMMER ONLY
DEG.PROG.:     M.S.
DEGREE:        B.S
UPDATED/BY:    4/25/76-MP

NAME:          JACKSON/JUDITH
ADVISOR:       COGDILL
AID:           RESEARCH ASST.
STATUS:        ON CAMPUS
DEG.PROG.:     M.S.
DEGREE:        B.S.
UPDATED/BY:    3/3/76-MP

NAME:          LECKNER/MICHAEL
ADVISOR:       MCILVAIN
AID:           TEACHING ASST.
STATUS:        ON CAMPUS
DEG.PROG.:     PH.D.
DEGREE:        M.S.
UPDATED/BY:    4/29/76-MP

NAME:          NORRIE/TERRI
ADVISOR:       HOOD
AID:           NONE
STATUS:        ON CAMPUS
DEG.PROG.:     M.S.
DEGREE:        B.A.
UPDATED/BY:    1/20/76-JK

NAME:          O'BRIEN/DON
ADVISOR:       SJOKEN
AID:           RESEARCH ASST.
STATUS:        ON CAMPUS
DEG.PROG.:     PH.D.
DEGREE:        M.S.
UPDATED/BY:    4/16/76-JK

NAME:          STORM/JEFF
ADVISOR:       DOZIER
AID:           NONE
STATUS:        IN ABSENTIA
DEG.PROG.:     ADV.CERT.
DEGREE:        M.ED.
UPDATED/BY:    2/28/76-JK

END OF CATALOG
```

----- APPENDIX C: HELP SECTION -----

```
CHOICES ARE:
  1) LOOK AT THE CATALOG -- YOU MAY JUST CHECK THE NAMES
     OF THE ENTRIES, LOOK AT ONE ENTRY, OR LOOK AT ALL OF
     THE ENTRIES IN THE CATALOG.
     CATEGORIES ARE:

NAME                          ADVISOR
AID                           STATUS
DEG.PROG.                     DEGREE
UPDATED/BY

  2) ADD AN ENTRY TO THE CATALOG
  3) ALTER AN EXISTING CATALOG ENTRY
       (NOTE: TO ALTER THE NAME OF AN ENTRY, YOU MUST
        DELETE THE ENTRY AND RE-ADD IT WITH THE NEW NAME)
  4) DELETE AN EXISTING ENTRY
  5) EXIT THE PROGRAM
  6) PRINT OUT THESE INSTRUCTIONS

IN QUESTIONS THAT REQUIRE A YES/NO ANSWER, 'Y' OR 'N' WILL
   SUFFICE.

WHENEVER YOU ARE ASKED TO GIVE THE NAME OF AN EXISTING ENTRY,
   YOU ARE ALLOWED TO USE ANY UNIQUE BEGINNING OF THE DESIRED
   NAME.  FOR EXAMPLE, FOR 'JOHNSON' YOU COULD TYPE 'J' IF NO
   OTHER ENTRY BEGINS WITH 'J'; HOWEVER, IF THERE WAS ALSO A
   'JONES', YOU WOULD TYPE 'JOH' FOR 'JOHNSON'.  IF 'JOH' WERE
   UNIQUE TO THE REST OF THE ENTRIES.  IF IT HAPPENS TO NOT
   BE UNIQUE, THERE WILL BE A MESSAGE TO THAT EFECT, AND YOU'LL
   GET ANOTHER CHANCE.
```

----- APPENDIX B: OUTPUT FROM INFOPR -----

GRADUATE STUDENT RECORDS

NAME	ADVISOR	AID	STATUS	DEG.PROG.	DEGREE	UPDATED/BY
BROWN/JAMES	DURBIN	NONE	ON CAMPUS	M.A.	B.S.	4/10/76-MP
COGSWELL/GAYLA	SJOKEN	NONE	SUMMER ONLY	M.S.	B.S	4/25/76-MP
JACKSON/JUDITH	COGDILL	RESEARCH ASST.	ON CAMPUS	M.S.	B.S.	3/3/76-MP
LECKNER/MICHAEL	MCILVAIN	TEACHING ASST.	ON CAMPUS	PH.D.	M.S.	4/29/76-MP
NORRIE/TERRI	HOOD	NONE	ON CAMPUS	M.S.	B.A.	1/20/76-JK
O'BRIEN/DON	SJOKEN	RESEARCH ASST.	ON CAMPUS	PH.D.	M.S.	4/16/76-JK
STORM/JEFF	DOZIER	NONE	IN ABSENTIA	ADV.CERT.	M.ED.	2/28/76-JK

APPENDIX D
LISTINGS

INFOS

```
10 REM INFORMATION STORAGE, MANIPULATION, AND RETRIEVAL SYSTEM
15 REM  1976   MARK W. PELCZARSKI      UNIVERSITY OF ILLINOIS
20 REM -- INITIALIZE FILES --
30 DIM S$(25)
40 FILE #4,"INFOLS"
50 QUOTE #4
60 PRINT "WHICH FILE DO YOU WANT? (TYPE 'NONE' FOR NONE)"
70 PRINT "CHOICES:"
80 RESTORE #4
90 IF END #4, THEN 130
100 READ #4,N$
110 PRINT N$;
120 GO TO 90
130 INPUT A$
140 IF A$="NONE" THEN 850
150 REM  - CHECK EXISTENCE OF CHOICE --
160 RESTORE #4
170 IF END #4, THEN 70
180 READ #4,N$
190 IF A$=N$ THEN 210
200 GO TO 170
210 FILE #3,A$
220 QUOTE #3
230 READ #3,P$,N$
240 REM -- CHECK PASSWORD --
250 IF P$="" THEN 320
260 PRINT "PASSWORD:";
270 INPUT M$
280 IF M$=P$ THEN 320
290 PRINT "TRY AGAIN, PASSWORD:";
300 INPUT M$
310 IF M$<>P$ THEN 60
320 LET P=VAL(N$)
330 LET M$="G.BAS $"+N$
340 FILE #1,A$+M$
350 FILE #2,A$+"F"
360 QUOTE ALL
370 RESTORE #2
380 SCRATCH #1
390 IF END #2, THEN 430
400 INPUT #2,M$
410 WRITE #1,M$
420 GO TO 390
430 FILE #5,"INFOSC.BAS $40"
440 REM -- HEADINGS FROM #3; S1 IS # OF HEADINGS --
450 READ #3,N$
460 PRINT "THIS IS A CATALOG FOR ";N$
470 FOR S1=1 TO 25
480 READ #3,S$(S1)
490 IF END #3, THEN 520
500 NEXT S1
510 REM -- INSTRUCTIONS --
520 PRINT "DO YOU WANT LONG FORM INSTRUCTIONS";
530 INPUT A$
540 IF A$="" THEN 520
550 IF LEFT$(A$,1)="Y" THEN 580
560 IF LEFT$(A$,1)<>"N" THEN 530
570 GO TO 720
580 GOSUB 2730
590 GO TO 720
600 REM -- LOOK AT THE CATALOG --
610 GOSUB 860
620 GOTO 720
630 REM -- ADDITION TO THE CATALOG --
640 GOSUB 1250
650 GOTO 720
660 REM -- ALTERATION OF AN ENTRY --
670 GOSUB 1730
680 GOTO 720
690 REM -- DELETE AN ENTRY --
700 GOSUB 2000
710 GOTO 720
720 PRINT "CHOICES: (1) CATALOG, (2) ADD, (3) ALTER, (4) DELETE, (5) EXIT,"
730 INPUT C
740 IF C<1 THEN 720
750 IF C>5 THEN 720
760 ON C GO TO 610,640,670,700,770,780
770 SCRATCH #2
780 RESTORE #1
790 IF END #1,THEN 830
800 READ #1,M$
810 PRINT #2,M$;
820 GO TO 790
830 SCRATCH #1
840 PRINT "FILES UPDATED -- EXIT"
850 CHAIN "ROUTER",100
860 REM -- LOOK AT THE CATALOG --
870 PRINT
880 PRINT "CHOICES: (1) LIST OF ENTRY NAMES, (2) DATA ON ONE ENTRY,"
890 PRINT "          (3) DATA ON ALL ENTRIES, (4) NONE"
900 INPUT C
910 IF C<1 THEN 880
920 IF C>4 THEN 880
930 ON C GO TO 950,1090,1190,1240
940 REM -- LIST OF NAMES --
950 PRINT
960 PRINT "**********"
970 PRINT
980 RESTORE #1
990 IF END #1,THEN 1040
1000 READ #1,N$
1010 PRINT N$
1020 SET #1,LOC(#1)+S1-1
1030 GO TO 990
1040 PRINT
1050 PRINT "**********"
1060 PRINT "END OF CATALOG"
1070 PRINT
1080 GO TO 880
1090 REM -- ONE ENTRY --
1100 PRINT "NAME OF ENTRY   ";
1110 INPUT N$
1120 GOSUB 2490
1130 IF L=0 THEN 880
1140 SET #1,L+1
1150 LET S=1
1160 GOSUB 2340
1170 GO TO 880
1180 REM -- ALL ENTRIES --
1190 RESTORE #1
1200 READ #1,M$
1210 LET S=0
1220 PRINT
1230 GOSUB 2340
1240 RETURN
1250 REM -- ADD AN ENTRY --
1260 PRINT
1270 RESTORE #1
1280 PRINT "DO NOT USE COMMAS IN ANY ANSWERS"
1290 PRINT "MAXIMUM LENGTH OF LINE SHOWN BELOW:"
1300 FOR C =1 TO P+1
1310 PRINT "-";
1320 NEXT C
1330 PRINT "!"
1340 PRINT S$(1)
1350 INPUT N$
1360 IF N$="" THEN 1350
1370 REM -- SEARCH FOR ALPHABETIZING --
1380 IF END #1,THEN 1470
1390 READ #1,M$
1400 IF N$<=M$ THEN 1430
1410 SET #1,LOC(#1)+S1-1
1420 GO TO 1380
1430 IF N$=M$ THEN 1460
1440 PRINT "AN ENTRY NAMED ";N$;" ALREADY EXISTS.  NAME MUST BE UNIQUE."
1450 RETURN
1460 SET #1,LOC(#1)-1
1470 REM -- INSERT --
1480 LET L=LOC(#1)
1490 IF END #1,THEN 1530
1500 READ #1,A$
1510 WRITE #5,A$
1520 GO TO 1490
1530 SET #1,L
1540 IF LEN(N$)=P THEN 1560
1550 LET N$=LEFT$(N$,P)
```

```
1560 WRITE :1,N$
1570 FOR N=2 TO S1
1580 PRINT S$(N)
1590 INPUT N$
1600 IF N$="" THEN 1630
1610 IF LEN(N$)<=F THEN 1630
1620 LET N$=LEFT$(N$,F)
1630 WRITE :1,N$
1640 NEXT N
1650 RESTORE :5
1660 IF END :5,THEN 1700
1670 READ :5,A$
1680 WRITE :1,A$
1690 GO TO 1660
1700 SCRATCH :5
1710 PRINT
1720 RETURN
1730 REM -- ALTER AN ENTRY --
1740 PRINT "WHICH ENTRY DO YOU WANT TO CHANGE ('NONE' FOR NONE)"
1750 INPUT N$
1760 IF N$="NONE" THEN 1980
1770 GOSUB 2490
1780 IF L=0 THEN 1740
1790 SET :1,L+1
1800 PRINT "AFTER EACH LINE, TYPE 'S' TO SAVE, 'C' TO CHANGE:"
1810 PRINT S$(1);":";M$
1820 FOR N=2 TO S1
1830 READ :1,M$
1840 PRINT S$(N);":";M$;"   ";
1850 INPUT A$
1860 IF A$="S" THEN 1970
1870 IF A$="C" THEN 1900
1880 PRINT "TYPE 'S' TO SAVE, 'C' TO CHANGE  ";
1890 GO TO 1850
1900 PRINT "CHANGE TO ";
1910 INPUT M$
1920 IF N$="" THEN 1950
1930 IF LEN(N$)<=F THEN 1950
1940 LET N$=LEFT$(N$,F)
1950 SET :1,LOC(:1)-1
1960 WRITE :1,M$
1970 NEXT N
1980 PRINT
1990 RETURN
2000 REM -- DELETE AN ENTRY FROM THE LIST --
2010 PRINT
2020 PRINT "WHICH ENTRY DO YOU WANT DELETED FROM LIST ('NONE' FOR NONE)"
2030 INPUT N$
2040 IF N$="NONE" THEN 2320
2050 GOSUB 2490
2060 IF L=0 THEN 2020
2070 PRINT "DELETE ";M$;" ? ('YES' OR 'NO')  ";
2080 INPUT A$
2090 IF A$="" THEN 2070
2100 IF LEFT$(A$,1)="Y" THEN 2140
2110 IF LEFT$(A$,1)<>"N" THEN 2070
2120 PRINT M$;" NOT DELETED"
2130 GO TO 2020
2140 RESTORE :1
2150 IF LOC(:1)=L THEN 2190
2160 READ :1,A$
2170 WRITE :5,A$
2180 GO TO 2150
2190 PRINT M$;" DELETED FROM LIST"
2200 SET :1,LOC(:1)+S1
2210 IF END :1,THEN 2250
2220 READ :1,A$
2230 WRITE :5,A$
2240 GO TO 2210
2250 SCRATCH :1
2260 RESTORE :5
2270 IF END :5,THEN 2310
2280 READ :5,M$
2290 WRITE :1,M$
2300 GO TO 2270
2310 SCRATCH :5
2320 PRINT
2330 RETURN
```



```
2340 REM -- PRINT DATA, IF S=1 THEN FOR ONE ENTRY, ELSE DUMP ALL --
2350 PRINT S$(1);":";M$
2360 FOR N=2 TO S1
2370 READ :1,N$
2380 PRINT S$(N);":";N$
2390 NEXT N
2400 IF S=1 THEN 2470
2410 IF END :1,THEN 2450
2420 READ :1,M$
2430 PRINT
2440 GO TO 2350
2450 PRINT
2460 PRINT "END OF CATALOG"
2470 PRINT
2480 RETURN
2490 REM -- SEARCH --
2500 IF A$<>"" THEN 2530
2510 INPUT A$
2520 GO TO 2500
2530 RESTORE :1
2540 LET L1=LEN(N$)
2550 IF L1=0 THEN 2610
2560 IF END :1,THEN 2610
2570 READ :1,M$
2580 IF LEFT$(M$,L1)=N$ THEN 2640
2590 SET :1,LOC(:1)+S1-1
2600 GO TO 2560
2610 PRINT N$;" WAS NOT FOUND"
2620 LET L=0
2630 RETURN
2640 LET L=LOC(:1)-1
2650 SET :1,L+S1
2660 IF LOC(:1)>LOF(:1) THEN 2690
2670 READ :1,A$
2680 IF LEFT$(A$,L1)=N$ THEN 2700
2690 RETURN
2700 PRINT N$;" NOT UNIQUE, MATCHES AT LEAST ";M$;" AND ";A$
2710 LET L=0
2720 RETURN
2730 PRINT
2740 PRINT "CHOICES ARE:"
2750 PRINT "  1) LOOK AT THE CATALOG -- YOU MAY JUST CHECK THE NAMES"
2760 PRINT "     OF THE ENTRIES, LOOK AT ONE ENTRY, OR LOOK AT ALL OF"
2770 PRINT "     THE ENTRIES IN THE CATALOG."
2780 PRINT "     CATEGORIES ARE:"
2790 PRINT
2800 LET N=1
2810 IF S1-N<1 THEN 2850
2820 PRINT S$(N),"  ",S$(N+1)
2830 LET N=N+2
2840 GO TO 2810
2850 IF S1-N<0 THEN 2870
2860 PRINT S$(N)
2870 PRINT
2880 PRINT "  2) ADD AN ENTRY TO THE CATALOG"
2890 PRINT "  3) ALTER AN EXISTING CATALOG ENTRY"
2900 PRINT "        (NOTE: TO ALTER THE NAME OF AN ENTRY, YOU MUST"
2910 PRINT "         DELETE THE ENTRY AND RE-ADD IT WITH THE NEW NAME)"
2920 PRINT "  4) DELETE AN EXISTING ENTRY"
2930 PRINT "  5) EXIT THE PROGRAM"
2940 PRINT "  6) PRINT OUT THESE INSTRUCTIONS"
2950 PRINT
2960 PRINT "IN QUESTIONS THAT REQUIRE A YES/NO ANSWER, 'Y' OR 'N' WILL"
2970 PRINT "   SUFFICE."
2980 PRINT
2990 PRINT "WHENEVER YOU ARE ASKED TO GIVE THE NAME OF AN EXISTING ENTRY,"
3000 PRINT "   YOU ARE ALLOWED TO USE ANY UNIQUE BEGINNING OF THE DESIRED"
3010 PRINT "   NAME.  FOR EXAMPLE, FOR 'JOHNSON' YOU COULD TYPE 'J' IF NO"
3020 PRINT "   OTHER ENTRY BEGINS WITH 'J'; HOWEVER, IF THERE WAS ALSO A"
3030 PRINT "   'JONES', YOU WOULD TYPE 'JOH' FOR 'JOHNSON', IF 'JOH' WAS"
3040 PRINT "   UNIQUE TO THE REST OF THE ENTRIES.  IF IT HAPPENS TO NOT"
3050 PRINT "   BE UNIQUE, THERE WILL BE A MESSAGE TO THAT EFFECT, AND YOU'LL"
3060 PRINT "   GET ANOTHER CHANCE."
3070 PRINT
3080 RETURN
3090 END
```

LISTING:
INPOPR

```
LIST
 10 REM INFORMATION STORAGE, MANIPULATION, AND RETRIEVAL SYSTEM
 15 REM   1976   MARK W. PELCZARSKI      UNIVERSITY OF ILLINOIS
 20 DIM S$(25)
 30 PRINT "      THIS PROGRAM SHOULD BE USED ON A DECWRITER II TERMINAL."
 40 PRINT "IF YOU WANT TO EXIT NOW, TYPE 'Y'; OTHERWISE TYPE 'N'."
 50 PRINT "EXIT";
 60 INPUT N$
 70 IF N$="Y" THEN 890
 80 MARGIN 132
 90 READ S
100 FILE #4,"INFOLS"
110 QUOTE #4
120 PRINT "WHICH FILE DO YOU WANT LISTED"
130 PRINT "CHOICES:"
140 RESTORE #4
150 IF END #4,THEN 190
160 READ #4,N$
170 PRINT N$,
180 GO TO 150
190 INPUT N$
200 IF N$="NONE" THEN 890
210 RESTORE #4
220 IF END #4, THEN 130
230 READ #4,A$
240 IF A$=N$ THEN 260
250 GO TO 220
260 FILE #3,N$
270 QUOTE #3
280 READ #3,M$,P$,A$
290 IF M$="" THEN 360
300 PRINT "PASSWORD";
310 INPUT B$
320 IF B$=M$ THEN 360
330 PRINT "TRY AGAIN, PASSWORD";
340 INPUT B$
350 IF B$<>M$ THEN 120
360 FILE #1,N$+"F"
370 FOR S1=1 TO 25
380 READ #3,S$(S1)
390 IF END #3,THEN 410
400 NEXT S1
410 LET P=VAL(P$)
420 LET N=INT(132/S1)
430 IF N<P+2 THEN 450
440 LET N=P+2
450 PRINT "HOW MANY COPIES DO YOU WANT";
460 INPUT C
470 IF C<0 THEN 450
480 IF C>50 THEN 450
490 PRINT "(1) SINGLE SPACED, OR (2) DOUBLE SPACED ";
500 INPUT A
510 IF A<1 THEN 490
520 IF A>2 THEN 490
530 PRINT "SET THE PAPER TO THE TOP OF THE NEXT PAGE NOW"
540 PRINT "THEN PRESS <RETURN>"
550 PRINT
560 PRINT
570 INPUT N$
580 FOR C1=1 TO C
590 PRINT
600 PRINT
610 PRINT A$
620 PRINT
630 PRINT
640 FOR I=1 TO S1
650 PRINT TAB(N*(I-1)+1);LEFT$(S$(I),N-1);
660 NEXT I
670 PRINT
680 PRINT
690 PRINT
700 LET L=8
710 RESTORE #1
720 IF END #1,THEN 820
730 FOR I=1 TO S1
740 INPUT #1,M$
750 PRINT TAB(N*(I-1)+1);LEFT$(M$,N-1);
760 NEXT I
770 FOR I=1 TO A
780 PRINT
790 NEXT I
800 LET L=L+A
810 GO TO 720
820 IF L<=S THEN 850
830 LET L=L-S
840 GO TO 820
850 FOR I=L+1 TO S
860 PRINT
870 NEXT I
880 NEXT C1
890 CHAIN "ROUTER",100
900 DATA 66
910 END
```


LISTING:
INPOAD

```
LIST
 10 REM INFORMATION STORAGE, MANIPULATION, AND RETRIEVAL SYSTEM
 15 REM   1976   MARK W. PELCZARSKI      UNIVERSITY OF ILLINOIS
 20 DIM S$(50)
 30 PRINT "WHICH FILE DO YOU WANT? (TYPE 'NONE' FOR NONE)"
 40 FILE #4,"INFOLS"
 50 PRINT "CHOICES:"
 60 RESTORE #4
 70 IF END #4,THEN 110
 80 READ #4,A$
 90 PRINT A$,
100 GO TO 70
110 INPUT F$
120 IF F$="NONE" THEN 1830
130 RESTORE #4
140 IF END #4,THEN 50
150 READ #4,A$
160 IF A$=F$ THEN 180
170 GO TO 140
180 FILE #3,F$
190 RESTORE #3
200 QUOTE #3
210 READ #3,P$,N$,A$
220 IF P$="" THEN 290
230 PRINT "PASSWORD";
240 INPUT B$
250 IF B$=P$ THEN 290
260 PRINT "TRY AGAIN, PASSWORD";
270 INPUT B$
280 IF B$<>P$ THEN 30
290 FOR S1=1 TO 50
300 READ #3,S$(S1)
310 IF END #3,THEN 330
320 NEXT S1
330 PRINT "CHOICES:  (1) CHANGE PASSWORD"
340 PRINT "          (2) ADD A NEW HEADING"
350 PRINT "          (3) CHANGE A HEADING (OR CHANGE ALL THE"
360 PRINT "              DATA UNDER A PARTICULAR HEADING, TO DO"
370 PRINT "              THIS, CHANGE THE HEADING TO ITSELF,"
380 PRINT "              THEN CHANGE THE DATA.)"
390 PRINT "          (4) DELETE A HEADING"
400 PRINT "          (5) EXIT THE PROGRAM."
410 INPUT C
420 IF C<1 THEN 340
430 IF C>5 THEN 340
440 ON C GO TO 450,560,850,1080,1830
450 REM -- CHANGE PASSWORD --
460 IF F$="" THEN 540
470 PRINT "NEW PASSWORD";
480 INPUT P$
490 PRINT "TYPE IT ONCE MORE";
500 INPUT B$
```

```
510 IF B$<>P$ THEN 470
520 GOSUB 1700
530 GO TO 330
540 PRINT "YOU CANNOT CHANGE A NON-EXISTENT PASSWORD"
550 GO TO 330
560 REM -- ADD A NEW HEADING --
570 PRINT "CHOICES:"
580 PRINT "  (1) INITIALIZE ALL DATA FOR THE NEW HEADING NOW"
590 PRINT "  (2) HAVE THE DATA FOR THE NEW HEADING SET BLANK"
600 PRINT "      AND INITIALIZE LATER"
610 INPUT C
620 IF C<1 THEN 570
630 IF C>2 THEN 570
640 GOSUB 1770
650 PRINT "TYPE THE NUMBER OF THE HEADING AFTER WHICH YOU WANT"
660 PRINT "  TO INSERT THE NEW HEADING (TYPE '0' TO NOT INSERT)"
670 INPUT C1
680 IF C1=0 THEN 330
690 IF C1<1 THEN 650
700 IF C1>S1 THEN 650
710 PRINT "WHAT IS THE NEW HEADING (IF YOU WANT A COMMA OR SEMICOLON"
720 PRINT "  IN IT, PLEASE ENCLOSE IN DOUBLE QUOTES)"
730 INPUT T$
740 FOR S2=C1+1 TO S1
750 LET V$=S$(S2)
760 LET S$(S2)=T$
770 LET T$=V$
780 NEXT S2
790 LET S1=S1+1
800 LET S$(S1)=T$
810 GOSUB 1700
820 LET S2=S1-1
830 GOSUB 1310
840 GO TO 330
850 REM -- CHANGE HEADING --
860 PRINT "CHOICES:"
870 PRINT "  (1) CHANGE CORRESPONDING DATA ALSO"
880 PRINT "  (2) LEAVE DATA ALONE"
890 INPUT C2
900 IF C2<1 THEN 860
910 IF C2>2 THEN 860
920 GOSUB 1770
930 PRINT "TYPE THE NUMBER OF THE HEADING YOU WANT TO CHANGE"
940 INPUT C1
950 IF C1<1 THEN 930
960 IF C1>S1 THEN 930
970 PRINT "WHAT IS THE NEW HEADING (IF YOU WANT A COMMA IN IT,"
980 PRINT "   PLEASE ENCLOSE THE HEADING IN DOUBLE QUOTES)"
990 INPUT T$
1000 LET S$(C1)=T$
1010 GOSUB 1700
1020 PRINT "NEW HEADING IS:",T$
1030 IF C2=2 THEN 330
1040 LET C=3
1050 LET S2=S1
1060 GOSUB 1310
1070 GO TO 330
1080 REM -- DELETE A HEADING --
1090 GOSUB 1770
1100 PRINT "TYPE THE NUMBER OF THE HEADING YOU WANT DELETED, OR"
1110 PRINT "  (TYPE '0' FOR NONE (DATA UNDER THAT HEADING WILL"
1120 PRINT "  BE LOST ALSO)   ";
1130 INPUT C1
1140 IF C1<1 THEN 330
1150 IF C1>S1 THEN 1100
1160 PRINT "DELETE:";S$(C1);"  ('YES' OR 'NO') ";
1170 INPUT T$
1180 IF T$="" THEN 1160
1190 IF LEFT$(T$,1)="N" THEN 1100
1200 IF LEFT$(T$,1)<>"Y" THEN 1160
1210 LET S1=S1-1
1220 FOR S2=C1 TO S1
1230 LET S$(S2)=S$(S2+1)
1240 NEXT S2
1250 PRINT "DELETED"
1260 GOSUB 1700
1270 LET C=4
1280 LET S2=S1+1
1290 GOSUB 1310
1300 GO TO 330
1310 REM -- FIX DATA --
1320 FILE #1,F$+"F"
1330 FILE #5,"INFOSC.BAS $40"
1340 RESTORE #1
1350 SCRATCH #5
1360 QUOTE ALL
1370 IF END #1,THEN 1410
1380 INPUT #1,T$
1390 WRITE #5,T$
1400 GO TO 1370
1410 RESTORE #5
1420 SCRATCH #1
1430 IF END #5,THEN 1670
1440 READ #5,V$
1450 LET T$=V$
1460 FOR I=2 TO C1
1470 PRINT #1,T$;
1480 READ #5,T$
1490 NEXT I
1500 ON C GO TO 1510,1560,1580,1620
1510 PRINT S$(1);":",V$
1520 PRINT S$(C1+1);
1530 INPUT V$
1540 PRINT #1,T$;V$;
1550 GO TO 1620
1560 PRINT #1,T$;"";
1570 GO TO 1620
1580 PRINT S$(1);":",V$
1590 PRINT S$(C1);
1600 INPUT T$
1610 PRINT #1,T$;
1620 FOR I=C1+1 TO S2
1630 READ #5,T$
1640 PRINT #1,T$;
1650 NEXT I
1660 GO TO 1430
1670 SCRATCH #5
1680 PRINT "DATA UPDATED"
1690 RETURN
1700 REM -- FIX-HEADING FILE --
1710 SCRATCH #3
1720 WRITE #3,P$,N$,A$
1730 FOR S2=1 TO S1
1740 WRITE #3,S$(S2)
1750 NEXT S2
1760 RETURN
1770 REM -- PRINT CURRENT HEADINGS --
1780 PRINT "HERE ARE THE CURRENT HEADINGS:"
1790 FOR S2=1 TO S1
1800 PRINT S2;") ";S$(S2)
1810 NEXT S2
1820 RETURN
1830 CHAIN "ROUTER",100
1840 END
```

INFO

A$ general variable used mostly for character string answers

C variable holding user's numerical choice

L location in data file of object of search

L1 the length of a data item

M$ general character variable

N loop index

N$ general character variable

P maximum data item length

P$ password

S switch for catalog output subroutine: S=1 for one entry, S=0 for all entries

S1 number of headings

S$(25) storage for headings

INFOPR

A switch for spacing: A=1 for single space, A=2 for double space

A$ general character variable

B$ used to check password

C number of copies

C1 index that counts copies

I loop index

L counts number of lines used on a page

M$ general character variable

N number of spaces per column

N$ general character variable

P maximum length of data item

P$ character equivalent of P

S lines per page as specified by DATA statement

S1 number of headings

S$(25) storage for headings

INFOAD

A$ file choices; description of file

B$ input password

C holds user's numerical choice

C1 number of heading that the user picks

C2 switch: whether corresponding data is to be changed

F$ data file name

I loop index

N$ character representation for maximum data length

P$ password

S1 number of headings

S2 loop index

S$(50) headings' storage

T$ general character variable

V$ general character variable

SE-933